I.J. Information Technology and Computer Science, **2016, 9, 39-48**
Published Online September 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.09.06

A Survey on Fault Tolerant Multi Agent System

Yasir Arfat
Department of Computer Science King Abdul Aziz University Jeddah, Saudi Arabia.
E-mail: yasirarfat081@gmail.com

Fathy Elbouraey Eassa
Department of Computer Science King Abdul Aziz University Jeddah, Saudi Arabia.
E-mail: fathy55@yahoo.com

***Abstract*—**A multi-agent system (MAS) is formed by a number of agents connected together to achieve the desired goals specified by the design. Usually in a multi agent system, agents work on behalf of a user to accomplish given goals. In MAS co-ordination, co-operation, negotiation and communication are important aspects to achieve fault tolerance in MAS. The multi-agent system is likely to fail in a distributed environment and as an outcome of such, the resources for MAS may not be available due to the failure of an agent, machine crashes, process failure, software failure, communication failure and/or hardware failure. Therefore, many researchers have proposed fault tolerance approaches to overcome the failure in MAS. So we have surveyed these approaches in this paper, whereby our contribution is threefold. Firstly, we have provided taxonomy of faults and techniques in MAS. Secondly, we have provided a qualitative comparison of existing fault tolerance approaches. Thirdly, we have provided an evaluation of existing fault tolerance techniques. Results show that most of the existing schemes are not very efficient, due to various reasons like high computation costs, costly replication and large communication overheads.

***Index Terms*—**Multi Agent System, Fault Tolerance, Agents, Adaptive Replication, Redundancy.

I. INTRODUCTION

A Multi-agent system (MAS) is composed of multiple interacting intelligent agents, within a given environment. These agents co-operate to solve difficulties that are beyond the capability or knowledge of each single problem solver. There are several key characteristics of agents, such as adaptation, scalability, re-usability, local view, autonomy, responsiveness and distribution. In order to achieve the necessary goals, agents are required to be able to communicate with many other agents in the environment ref. Byrski et al.[1] There are various applications of MAS like aircraft maintenance, environment monitoring, military demining, surveillance, internet agent, health care, spacecraft control and industrial monitoring [2][3][4].

In this paper, we focused on the fault tolerance of MAS. There are various fault tolerance needs that MAS should be contained in, in order to mitigate failure.

- Agents need to collaborate with each other to avoid failure
- Information sent over MAS should be transparent during transmission
- Availability of other agents in MAS should be ensured when an agent fails
- The agent's system should have the ability to take decisions based on knowledge
- Agents need to communicate in a secure manner and data should be protected in case of failure
- Agents should have autonomy in case of failure. They should be able to provide services without affecting other agents
- An agent should have scalability and complexity so that it can deal with any size of agent without affecting performance
- An agent should be robust enough to confront any failure i.e. process failure, crashing failure etc. whereby it can provide services without any interruption.
- Agents should have the ability to adapt to any condition, in any environment, in case of failure.

In MAS, there are several factors that decrease performance and reliability. One of these is a failure of the system. If there is any fault in the system, it will stop working and cause a delay in achieving the required goals. In order to increase the reliability of MAS, the system should be fault tolerant. If there is any fault in the system it should have the ability to mask the failure in order to continue providing the necessary services without any delay ref. Abbas et al.[5]. If a system is fault tolerant then it will also increase the performance of the system.

In this research paper, we classify the fault tolerant multi-agent system (FTMAS) into different categories, on the basis of recovery techniques and by presenting the taxonomy of both faults and techniques. Also, we have provided a qualitative comparison of the recent fault recovery of MAS. We discovered that researchers are applying replication and non replication based fault recovery approaches for FTMAS. We also examined existing techniques on the basis of their attributes such as, characteristics, failures, types of agents, environment and replication protocol.

The rest of the paper is organized as follows. Section II contains a brief background of FTMAS, Section III briefly describes a summary of existing literature. Section IV presents a taxonomy of FTMAS. In section V, we have provided a comparison of FTMAS with other techniques. Section VI summarizes the advantages and disadvantages of existing techniques in FTMAS. In section VII, we conducted an evaluation. In section VIII, we have discussed future challenges and issues. Finally, section IX concludes the paper.

II. Background

Multi agent system comprises of various agents, entities etc. A single agent is capable of carrying out independent actions to achieve the delegated goals. This agent system may work in different environments according to the tasks set to it, responsibilities assigned to it by the system or program inside the agent system. Figure1 shows the basic operation and work performed by the agent system

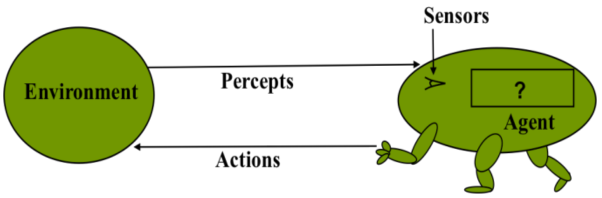

Fig.1. A multi agent system

A specific agent will carry out specific goals in the multi agent system, for example: *Environment*: Patient & hospital, *Goal*: Healthy patients, *Actions*: Tests and treatments, *Percepts***:** Patient symptoms, *Agent Type***:** Medical diagnosis. An agent has a range of characteristics i.e. reactiveness, reliability, scalability, autonomy, robustness, intelligence, persistency, goal-orientation, adaptability and sociability. These are the basic characteristics of an agent that it contains.

In multi agent systems (MAS), several agents are working together to achieve task-oriented goals on behalf of the user or human ref. Maciel et al.[6] Successful interaction is required among agents in MAS to negotiate, coordinate and cooperate with each agent in the environment ref. Gerrard et al.[7] The best examples of MAS are Internet agents and as used in Spacecraft control ref. Li et al.[8] Nowadays, researchers and developers alike are using the agent in the distributed environment, such as those used as environment agents who need co-ordination, co-operation, and negotiation. These are the basic issues that MAS has in each environment ref. Davoodi et al.[9]. As the failure rate increases when there is less co-ordination, co-operation and communication among the agents, this leads to the failure of the system. Hence, these types of failures are subject to the host, machine and exception set ref. Wang al.[10].

There are several fault tolerance MAS techniques that have been proposed to mask the faults in MAS. Each technique differs in its ability to mask failure in MAS.

III. Literature Review

This section presents an overview of the state-of-the-art of fault tolerance techniques in the multi agent system (FTMAS). This overview comprises of discussion about the assumptions, objectives, methodologies and key approaches present in existing works. Based on this section, we will present a taxonomy and comparison in the following sections.

A. Towards FTMAS Architecture

Kumar et al. [11] describe that there were many possibilities that failure could happen at any time in MAS of any distributed system. Many agents were not available due to process failure, exceptions and breakdown of communication. There were many faults that existed ranging from database recovery, TS monitoring, resource manager and fault tolerance distributive systems up to application server. There were many issues in these techniques, such as using replication schemes as a critical system for monitoring. However, when it increases the reliability of the system it duplicates the data and services. Moreover, many systems saved the application state but it also created many problems during recovery. To overcome this traditional fault tolerance technique they proposed Adaptive Agent Architecture (AAA) for the multi-agent system (MAS). Whereby, AAA overcomes a problem like a broker failure without incurring undue overheads. There may be more than one of many such brokers in the large multi-agent system. In the case of sudden unavailability of a broker in AAA, they used the team based approach for automatic recovery of MAS. Furthermore for the recovery, they assumed three different recovery schemes, namely logical characterization, recovery scheme and recovery scenario. In these assumptions, they described different steps, theorems and characterizations of performance. Their results show that autonomous agents can make a multi-agent systemmore robust.

B. Towards Adaptive FTDMAS

Marin et al. [12] have also proposed an adaptive architecture for the multi-agent system (MAS). It deals with existing problems in MAS using new methodologies. MAS as a distributed system may by its very nature accrue failure at any time in the system. Moreover, due to it being a distributed system, computations of dynamic applications were often changed, during execution. Nevertheless, they tried to make it more flexible to overcome the flaws of the conventional system. On the proposed architecture, we can either replicate or replicate the software element on the spot. The advantage of this approach is that we can change replication tactics in a matter of a few seconds. The main objective of selected architecture is to make fault tolerance more efficient for MAS, using selective replication techniques. An outcome of this approach is to develop architecture, which is suitable for dynamic fault tolerance for applications. They used the selective replication scheme as many problems existed for approaches to dynamic applications. Moreover, they also introduced a framework namely, (dynamic adaptive replication extension) DARX, which uses both active and passive replication, specially designed for the distributed application. It has many advantages i.e. to dynamically add or remove replicas, atomic and ordered multicast for each replication group etc. To manage the failure of the system, there is a replication manager associated with each group which performs the following functions: 1)Maintenance of information within the group 2)Perform suspension and resumption activity 3)Diffusion of a message and 4)Switching the replication strategy. The benefits of performing these functions are that: the replication manager can recover the failure quickly; when one group fails, the other groups have all the information needed in order to active a new replica. A simulation display ensures that minimum energy is utilized between nodes to carry out the task, as a single copy of data will be sent and it also improves the probability of delivery.

C. Towards Automatic FTMAS

Almeida et al. [13] presented an automated fault tolerance (FT) MAS scenario. They described that there are many possibilities whereby an exception or failure can occur at any time in the system. These failures occur when recovery and fault tolerance approaches are defined at the design level. Indeed, it is very difficult to decide at the design level when and where to apply the FT approach (i.e. replication). But conventional approaches are out of order when it comes to dynamic systems (i.e. MAS). These applications could be ambient intelligence systems, related to e-commerce, crisis management systems or the air traffic control system. According to the situation and nature of interdependencies in these applications, an agent can change their role during the computation stage. Therefore, to overcome all difficulties and to make the FT management automatic and dynamic, they considered a self-adaptation FT approach.

In MAS, multiple errors may occur as they are only considered as crash type failures. Thus, to mask these types of failures, replication is considered an ideal approach. There are various types of replication approaches, from static to non-static and explicit replication. Moreover, for replication they presented dynamic and automatic control of replication. Hence, they chose a DARX framework, which has dynamic distributed replication features. Using this system, they have estimated the critical essence of the system by concluding with different types of information i.e. messages, plans and roles etc.

D. Decentralized Architecture for FTMAS

Khan et al. [14] presented fault tolerant decentralized architecture for the multi-agent system. Most applications have a lack of fault tolerance. There is an expectation that usage of MAS in different distributed applications will increase. However, there are many faults existing within the agent platform, causing a multitude of problems. To overcome all these problems they introduced decentralized architecture, as an alternative to the centralized architecture of the agent platform (AP). Figure 2 shows the working of decentralized architecture, namely Virtual Agent Cluster (VAC). When a single agent platform is deployed it includes all machines. A similarity exists between virtual agent cluster and cluster computing, where the front processor distributes the load among the machines.

Agent Communication Language (ACL) also acts as a front processor. It is used as an interface to communicate with another agent in the system. The communication between machines is bi-directional through IP addresses, whereby in each machine there is an Agent Management System (AMS). It is organized in such a manner that failure of one machine does not affect the other. There are several characteristics of this architecture, which includes fault tolerance and recovery, autonomy, application layering architecture (inter VAC and intra VAC) and load balancing. These characteristics make this decentralized architecture more flexible in scale and fault tolerant. Moreover, fault tolerance is the greatest substantial advantage that can be achieved through this decentralized architecture.

E. Plane-Based Replication for FTMAS

Almeida et al. [15] presented a plane-based replication of the fault tolerant multi-agent system. In their proposed scheme, they used this method for stipulating the dependability for MAS through replication. This method is different from others cited above, here they focus on predictive and adaptive replication whereby the critical agents are replicated to overcome failures. As some of the application uses static replication, in contrast here they use dynamic replication. The latter has advantages over static replication i.e. re-allocation of tasks, changing the role of an agent, flexible organization etc. Moreover, it is very important to replicate an agent through dynamic and automatic means. Here, they are more focused on building reliable MAS. Hence, a plan based fault tolerance method promising prevention because it predicts upcoming behavioral patterns of an agent. To make MAS more reliable, original predictive approach calculates criticality of the agent dynamically. Then this criticality of the agent is used for replication, in a manner to increase the dependent ability on the basis of resources that are available. They also validated their approach on the DARX framework and DIMA. In this strategy, an agent is accomplished as DIMA agent and usage of DARX in command is used to obtain replication capabilities.

F. Adaptive and Automated FTMAS

Singh et al. [16] have proposed this framework for a critical agent in the multi-agent system (MAS), based on the cardinality of an agent. Sometimes replication can become very costly due to the complexity of the system; moreover, dynamic replication is also a need of all agents in fault tolerance MAS. Hence, to overcome these issues they proposed this particular framework. They mixed two techniques namely, active and passive replication. Thereby, critical agents will actively replicate, more focused relatively to other agents. The benefit of this approach is to reduce the complexity of the system, cost, optimal utilization and more importantly, optimal fault tolerance. The proposed framework is hybrid, having the automated and adaptive characteristics of fault tolerance. This framework has three different components: 1) Replica Store 2) Fault Management Agent (FMA) and 3) Event Monitoring Agent (EMA).

In replica store central fault management unit (CFMU) divides it into two phases, active and passive replica store. A passive replica is used to update the agent periodically. For critical agents, active replica are used to having a working replica. All faults management control is done by the FMA. It also retains information about the replication, whether it is active or passive replication. The last component is EMA, which is responsible for keeping track of the information related to crashes, putting in the substitute replica for that agent. Results show that fifty percent actively replicated agent can remove the complexity of the system. Moreover, from the proposed system, scalability of the fault tolerance multi-agent systems can be improved.

G. Hybrid Based Approach for FTMAS

Koppensteiner et al. [17] have proposed a hybrid fault tolerance multi-agent system using the heartbeat mechanism. They used this mechanism to detect failure in MAS. They found three different types of failures here, namely: 1) System disturbance 2) Physical Component Failure and 3) Software Entity Failure. To recover from physical component failure i.e. a failure in tangible hardware or failure in block base application controlling function, they introduced the heartbeat mechanism. Using the heartbeat between the LLC (Low-Level Control layer) and HLC (High-Level Control layer) they minimized messages to maintain the system's stability. This approach also implements the heartbeat method exclusive from distribution of messages on the system. If there is a fault inside the system, they can only communicate messages if necessary. In a situation of complete failure in the system, both LLC and HLC will be used to detect which agent has failed. Utilizing the heartbeat method, they will try to fix it.

H. Choice of Sampling Rates in FTMAS

Bora et al. [18] proposed fault tolerance in a multi-agent system based on the sampling period. To increase the fault tolerance in distributed and dynamic systems, adaptive replication techniques were very useful. But there is one disadvantage of this approach; it increases the cost due to adaptive replication. To overcome this drawback, a sampling period was introduced to minimize the cost. This technique whereby it monitors critical agents, properly chooses the appropriate replication for the agent based on its criticality. They applied this technique on abstract architecture for adaptive replication. This architecture consists of replication manager; which is responsible for providing active and passive replication among different replica. It also monitors and handles faults inside the replica. The Main modules contained in this architecture are observation and feedback control. The replication manager utilizes these features. The Observation module collects information about the system and passes this information to feedback control. All this information is processed by feedback control, which then decides which agent is most critical, having calculated their relative critical value. Then it applies the adaptive replication policy based on the criticality of the system. This architecture covers the crash type of failure in multi-agent systems. In this research paper, they also assume that the sampling period will maximize accuracy, reduce the cost of replication and increase the response time of the system.

I. A Decision-making based approach for Fault Handing in Multi-Agent System

Mirian, Maryam S. et al. [19] introduced a new decision-based technique for fault handling in the multi-agent system. They described the multi-agent system more like a distributed system where fault can occur at any time in the system. In the paper, they focused on the faulty agent and their recovery in the multi-agent system. In the presented technique, if a fault agent requests its other agents or its team agents come to know that this agent is faulty and needs help, then there are several help requests that exist. However, which help request is appropriate and which will be effective are all decided at the decision-making phase. At this stage, they also use the best fit, first come first serve and shortest job first algorithm to making the decision for the help request. For this methodology there is no central agent, all agents are decentralized. Each agent has knowledge about the environment and existing agents in the environment. They all also have the ability to perform the task of other agents. If an agent fails in the system another agent can help based on the decision-making phase.

J. Distributed Adaptive Fault-Tolerant Consensus Control of Multi-Agent System with Actuator Faults

Khalili et al. [20] presented a distributed FT consensus control of MAS with actuator faults. This FTMAS is based on three different assumptions. In this distributed system an FT control component was developed to perform a two-step process between the agents. The first would diagnose the fault in the MAS while the second would provide an opportunity to recover in an adaptive manner. These assumptions are constructed using mathematical equations and in particular, vectors. Using the assumptions, it can check the system's stability with the closed-loop mechanism. The main objective of this system is to develop an algorithm that diagnoses and recovers faults. A unique feature of this algorithm is that it takes an information-neighboring algorithm and applies its actions.

IV. TAXONOMY OF FAULTS IN MAS

In this section, we have presented a taxonomy of faults and their related techniques. First of all, we divided the faults into two different categories, namely fail silent and fail uncontrolled. These are the faults we found in a different paper that we surveyed. Fail silent faults are those, which belong to the crash type of failure. On the other hand fail uncontrolled are those failures, where any type of fault or failure can occur. The faults are then further subdivided into different types as given in Figure 2.

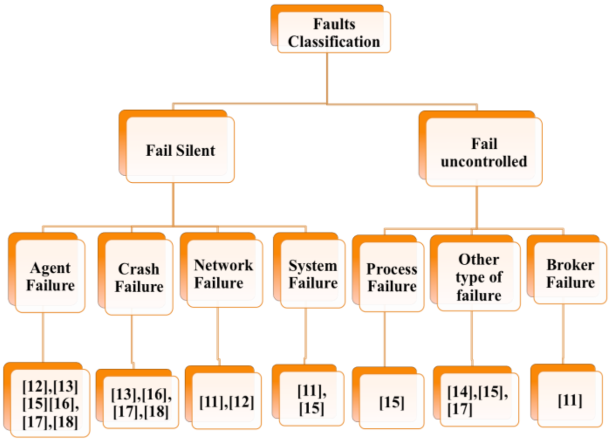

Fig.2. Taxonomy of faults in MAS

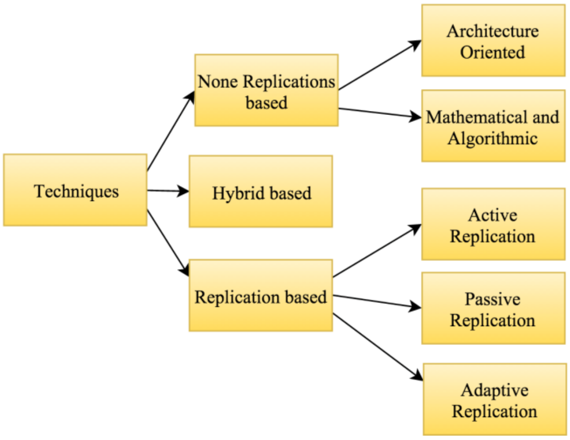

Fig.3. Taxonomy of techniques against the faults in MAS

We also found out the taxonomy of the techniques that researchers are applying for fault tolerance in the multi agent system. We have classified these approaches into three different categories. These are replication based, non-replication based and hybrid approaches. Then we further subdivided these techniques, for example the replication based approach has active replication, passive replication and adaptive replication. Moreover we also further subdivided the non-replication based approaches into two different types, these being architecture oriented and mathematical/algorithmic, which are given in Figure 3. These are the existing techniques that are used for fault tolerance, if there is any fault in the system using these techniques we can avoid failure of the whole system. These approaches have their own advantages and disadvantages, which vary according to the environment where these methods are being applied.

V. QUALITATIVE COMPARISON OF FTMAS

In this section, we have provided a qualitative comparison of the existing fault tolerance approaches of MAS as given in table 1. For this purpose we have used the following parameters: 1) Agent type 2) Fault tolerance technique 3) Objectives 4) Language 5) Type of failure 6) Replication protocol 7) Characteristics and 8) Environment.

According to this table, Kumar et al. [11] have adopted the object replication based fault tolerance approach for MAS, which has characteristics like autonomy and local view, whereby the main objectives of this approach achieve a faster fault recovery as they use the broker process failure. Moreover, Marin et al. [12] and Almeida et al. [15] provide the dynamic replication approach for the fault tolerance multi-agent system, having the objective to achieve that the agent should execute the goal. Furthermore, this technique covers machine and host failures, network failure and distributed agent failures. They are also using different types of agents, namely selective agent and critical agent. Moreover, both are using the Knowledge Query Manipulation Language (KQML) for agent communication among each other.

Table 1. Qualitative Comparison

Research Paper	Technique	Characteristics	Type of Failure	Replication Protocols	Objectives	Agent Type	Environment
Kumar et al. [11]	Adaptive Agent Architecture	Autonomy, Local views	i) Machine Crashes. ii) End of broker process. iii) Network Break Down	Object group replication	i) To achieve warm backups. ii) Object group and virtual Synchrony	Complex Agents	Virtual Environment
Marin et al. [12]	Bypass dynamic Replication.	Autonomy, Run time replication change	i) Host and Network failures. ii) Failure of an agent in distributed applications	Dynamic Replication	i) Efficient FT for MAS through Selective Agent Replication ii) Appropriate MA architecture for dynamic FT	Selective Agent	Continuous Environment
Almeida et al. [13]	Self adaptation of fault tolerance	Autonomous, Dynamic, automatic	i) Crash type of failure cause by the internal (hardware	Adaptive Replication	i) Make autonomous the Management of fault tolerance,	Critical Agent	Discrete Environment

			issue and OS crashes) or external factor (malicious attacks, environment tragedy and power failure)		ii) To make this fault tolerance management dynamic and automatic		
Khan et al. [14]	Virtual Agent Cluster	Faster Recovery and Fault Tolerant, Autonomous, Architecture for application layering (intraVAC and interVAC), Balancing the Load	i) Centralized AMS lack of fault tolerance. ii) Centralized system become bottleneck under heavy load. iii) Utilization of information service and provision QOS (timelines and reliability)	Active replication	i) To embrace peer-to-peer computing paradigm. ii) Eliminate the limitations in present in existing Agent Platform (AP)	Active Agent	Virtual Environment
Almeida et al. [15]	Replication of critical agents	Predictive, Automatic and Adaptive	i) An agent failure or a machine. ii) Host failure and process failure base on adaptive failure indicators hierarchy.	Active and Passive Replication, (Dynamically apply)	i) Provide some action based on after calculation of value of a system that will be executed by the agent near future in case of failure of an agent. ii) Each agent should execute each plan (action) in order to achieve the goal.	Critical Agent	Continuous Environment
Singh et al. [16]	Automatic and adaptive fault recovery, Central fault Management	Local views	i) Crash or failure of an agent. ii) Critical agent failure	Adaptive Replication	i) Using the both active and passive replication make the MAS more scalable, reliable and fault tolerant. ii) To reduce cost and complexity of the system.	Critical Agent	Transient Environment
Koppensteir et al. [17]	Heartbeat mechanism (failure detection), Supervisor Agent Approach (system failure absorption, fault recovery)	Autonomous, Local views	i) Physical Components failure (Breakdown of whole resource, temporary failure) ii) Software entity failure iii) Partially agent failures		i) To increase the stability of the system To shorter the reaction time. ii) To enhance the fault tolerance of a complex system	Complex Agent	Discrete Environment
Khalili et al. [20]	Distributed adaptive fault-tolerant	Autonomous, Adaptive.	i) Crash failure ii) failure of an agents	Adaptive Replication	To reduce the cost of replication in MAS and computation overhead.	Critical Agent	Discrete Environment

Additionally, Almeida et al. [13] and Khan et al. [14] also presented a replication and non-replication based approach respectively. To overcome the drawback of the Khan et al. [14] centralized MAS system, they presented decentralized architecture whereby centralized architecture was thought to develop bottlenecks under a heavy load. They have used the Agent Communication Language (ACL) for communication between agents by using active agents. The main objectives of this technique are i) To embrace a peer-to-peer computing paradigm and

ii) To eliminate the limitations present in existing Agent Platform (AP).

On the other hand Almeida et al. [13] have provided KQML for the agent. They also used a discrete environment where the agents can perform only limited actions. The goal of this approach is to make autonomous management of fault tolerance and also make this fault tolerance management dynamic and automatic.

Singh, Aarti et al. [16] have provided automatic and adaptive fault recovery for the multi-agent system. Characteristics of these approaches are a local view and autonomy. The main objective of this approach is to achieve i) using both active and passive replication to make MAS more scalable, reliable and fault tolerant. ii) To reduce cost and complexity of the system. They are also using the agent in a transient environment. The technique that they provided covers the following types of failures:

i) Crash or failure of an agent ii) Critical agent failure.

Koppensteiner et al. [17] and Khalili et al. [18] have presented the heartbeat mechanism and choice of the sampling rate for fault recovery respectively. On the other hand these approaches cover physical components failure, partial agent failure, crash failure and failure of agents.

This comparison gives us a clear idea of different approaches for fault tolerant multi-agent systems and how to deal with these failures using an appropriate approach, which is efficient, with less overheads, easy to use and less costly.

Table 2. Pros and Cons of Existing Techniques

Research Paper	Technique	Pros	Cons
Kumar et al. [11]	Adaptive Agent Architecture	i) Effects of recovery on response time. ii) Effects of transition on response time. iii) Less overheads of using teamwork.	i) More focused on broker failure tolerance. ii) Less focused on individual agents. iii) Require extra computing for the management of brokerage layer.
Marin et al. [12]	Bypass dynamic Replication	i) Fast way to handle faults and recovery. ii) Improve reliability, fault-tolerance. iii) Improve accessibility.	i) Replication is very costly ii) More Computations are required.
Almeida et al. [13]	Self adaptation of fault tolerance (Dynamic adaptation of replication strategies)	i) It provides dynamic replication; we can use both active replication and passive replication. It provides better recovery.	i) When we use both active replication and passive replication in MAS, such an environment proves very costly.
Khan et al. [14]	Virtual Agent Cluster	i) Decartelized MAS is less faulty as compared to centralized. ii) More reliable. iii) Faster as compared to centralized.	i) It falls short of addressing the heterogeneity issue. ii) Cost implication of recovery in multi organizational context. iii) More overheads.
Almeida et al. [15]	Replication of critical agents	i) It provides replication for the criticality of agent, which is more critical than applying the replication for them.	i) It is very hard to find out which agent is more critical in multi agent system for fault tolerance.
Singh et al. [16]	Automatic and adaptive fault recovery, central fault Management	i) Using this approach, it provides automatic recovery for faults when they occur and adaptive fault recovery.	i) This approach has high overheads. ii) Reliability of this technique is less as compared to the other listed techniques in this table.
Koppensteiner et al. [17]	Heartbeat mechanism (failure detection) and Supervisor agent approach (system failure absorption, fault recovery)	i) It provides faster fault recovery. ii) Sending messages at a specified period of time through which it can easily find out which agent has failed, thus providing quick recovery.	i) Computation cost is high as it involves a lot of work communicating with each agent. ii) Reliability is very low. iii) Very slow technique that causes more overheads due to sending regular messages after a specified period of time.
Bora et al. [18]	Adaptive Replication based on sampling rates.	Using the sampling rate replication cost will decrease. Response time of switching from active to passive and visa versa will decrease. Fault and reliability can be achieved easily using the sampling rate. Adaptive replication increases the response time of the system.	i) Adaptive Replication is very costly. ii) Overhead will be very high.

VI. PROS AND CONS OF EXISTING TECHNIQUES

Here in this section we have described the different approaches and their pros & cons using different parameters as given above [11], [12], [13], [14], [15], [16], [17],[18],[19],[20]. In this section we find out the advantages and disadvantages of the fault tolerance approaches that we surveyed in the literature review. In given table 2 we can see that there are some techniques that are providing better fault tolerance recovery for any failure of the multi agent system. This shows us how much the techniques are effective for fault recovery. When the researcher proposed a technique for fault tolerance they ignored other aspects of fault tolerance, as it can cause high overheads and perform some expensive computations. They also decreased the reliability of the system and reduced the performance of MAS. Moreover, table 2 shows overheads of fault recovery, reliability, improvement in performance and computational cost of these approaches.

VII. Discussion of Evaluations and Comparison

In this section, we evaluated different schemes and accessed them for fault tolerance recovery. Beginning with Kumar et al. [11] who proposed adaptive agent architecture to mask the failure in the multi-agent system. This has several characteristics namely, autonomy, local view and mobility. Using this approach they have covered different types of failure namely, machine crashes, end of broker process and network bread down. Moreover, they applied the object group replication for redundancy in MAS. The main objective of this approach is to achieve warm backup, object group and virtual synchrony. Marin, Olivier et al. [12] have proposed the dynamic replication technique. Characteristics of this approach are Autonomy and it has run time replication changes. Using this technique they mask the failure of host and network, thus effectively the failure of any agent in the distributed environment. They applied dynamic replication protocol for redundancy. The main objective of this approach provides efficient FT for MAS through selective agent replication. Almeida et al. [13] have applied self-adaptation of fault tolerance approach having the characteristics of dynamic and automatic recovery of fault. They tried to mask failures such as: crashes caused by internal hardware issues and operating system crashes or external malicious attacks, environmental tragedy and power failure. They have applied adaptive replication. The main objective of this technique is to make autonomous the management of fault tolerance and also to make this fault tolerance management dynamic and automatic. Khan, Abbas et al. [14] have virtual agent cluster (VAC) with the following characteristics: Faster recovery and fault tolerant, autonomous, architecture for application layering into intraVAC and interVAC and balancing the load. They tried to mask failures such as crashes caused by internal hardware issues and OS crashes or external malicious attacks, environmental tragedy and power failure using the active replication. The main objective of this scheme is to embrace a peer-to-peer computing paradigm and eliminate the limitations present in existing agent platform (AP). For communication of agents with other agents in MAS, they have used Agent Communication Language (ACL). Almeida, Aknine et al. [15] have provided replication of critical agents having features namely, predictive, automatic and adaptive. They tried to mask agent or machine failure. Host failure and process failure are based on an adaptive failure indicators hierarchy. Their main objective was that an agent should execute each plan (of action) in order to achieve the overall goal.

Singh et al. [16] chose an automatic and adaptive fault recovery and central fault management technique. They tried to minimize system crashes, agent failures and critical agent failure. Their main objectives are to reduce cost and complexity of the system. Meanwhile, Koppensteiner et al. [17] have implemented the heartbeat mechanism for failure detection, supervisor agent approach for system failure absorption and fault recovery. They applied this technique to overcome physical components failure and breakdown of the whole resource including temporary failure. The main goal is to increase the stability of the system and shorten the reaction time - overall - to enhance the fault tolerance of a complex system. In Khalili, Mohsen et al. [20] they have proposed a sampling rate in FTMAS which tries to overcome crash failures or failures of an agent in the MAS environment, whereby they applied adaptive replication for redundancy to mask the failure in FTMAS. This technique has performed relatively better than other techniques but it also has large overheads of FT for recovery.

Moreover, we have seen that some approaches perform better in fault recovery as compared to other techniques. Some techniques have higher overheads and perform costly computations for fault tolerance. There should be an appropriate technique that enhances performance on fault recovery, rather than making it suffer.

VIII. Future Challenges and Issues

There has been advancement in multi-agent system technology and its usage in our daily life is increasing. Even though a lot of work has been done for fault tolerance in a multi-agent system (FTMAS) but the issue regarding failure recovery of MAS has still not been overcome yet. As MAS gets further distributed, failure can occur at any time.

There are various challenges in FTMAS implementation. It is still a complex task.

- From the literature survey, we found that most of the existing fault tolerance approaches are not providing basic fault recovery features in MAS like reliability, scalability, adaptability and robustness.
- A challenging issue in designing fault tolerance architecture for Multi Agent system (MAS) is its distributed nature, prone to failure at any time.
- Another major problem is that there is no standard evaluation for the framework of FTMAS that is needed for comparison purposes. Currently each researcher uses their own criteria for evaluation.
- MAS has a lack of reliability in programming tools and specialized debugging tools. Skills are also needed to shift from an analysis and design phase to coding, as well as issues in understanding the environment and methodology.

IX. Conclusion

Currently, multi-agent system is being used in different applications in a distributed environment. In MAS, as there are many agents so there are several challenges that can occur. For example, co-ordination, co-operation, negotiation and communication in a distributed environment. When one agent does not co-operate due to a fault then other components of MAS also do not provide their services. Then failures like machine crashes, process failure, software failure, communication failure and hardware failure occur. Therefore, in this research paper, we have surveyed the many techniques for fault tolerance in a multi-agent system so that failures can be overcome. In this research paper, we have presented existing techniques, which are very effective for fault tolerance, by providing related work and then classifying these approaches into different categories. We also categorized failures that occur in the multi agent system. Furthermore, we have also provided a qualitative comparison of existing fault tolerance approaches. In this comparison, we locate different parameters so that we can identify by comparing which technique is better for masking a fault in MAS. We have provided the pros and cons of existing fault tolerance techniques. It shows that most of the existing schemes are not efficient due to various reasons like high computation cost, costly replication and large communication overheads. We have found out that when researchers proposed a technique for fault tolerance, they ignored its overheads which when applied to MAS, proved very costly. It provides fault tolerance but on the other hand, it also degrades the performance of the systemand reliability. There should be an appropriate technique, which provides fault tolerance with fewer overheads and hence less expensive for computation.

References

[1] Byrski, Aleksander, Rafał Dreżewski, Leszek Siwik, and Marek Kisiel-Dorohinicki. "Evolutionary multi-agent systems." The Knowledge Engineering Review 30, no. 02 (2015): 171-186.

[2] Eddy, Foo, H. B. Gooi, and S. X. Chen. "Multi-Agent System for Distributed Management of Microgrids." Power Systems, IEEE Transactions on 30, no. 1 (2015): 24-34.

[3] Sajja, Priti Srinivas. "Automatic Generation of Agents using Reusable Soft Computing Code Libraries to develop Multi Agent System for Healthcare."International Journal of Information Technology and Computer Science (IJITCS) 7, no. 5 (2015): 48.

[4] Yadav, Sandeep Singh, and Mandeep Singh Yadav. "Development of System for Automated & Secure Generation of Content (ASCGS)."International Journal of Information Technology and Computer Science (IJITCS) 7, no. 11 (2015): 81.

[5] Abbas, Hosny Ahmed, Samir Ibrahim Shaheen, and Mohammed Hussein Amin. "Organization of multi-agent systems: an overview." arXiv preprint arXiv:1506.09032 (2015).

[6] Maciel, Cristiano, Patricia Cristiane de Souza, José Viterbo, Fabiana Freitas Mendes, and Amal El Fallah Seghrouchni. "A Multi-agent Architecture to Support Ubiquitous Applications in Smart Environments." In Agent Technology for Intelligent Mobile Services and Smart Societies, pp. 106-116. Springer Berlin Heidelberg, 2015.

[7] Gerrard, Claire E., John McCall, Christopher Macleod, and George M. Coghill. "Applications and design of cooperative multi-agent ARN-based systems." Soft Computing (2015): 1-14.

[8] Li, Ni, Xiang Li, Yuzhong Shen, Zhuming Bi, and Minghui Sun. "Risk assessment model based on multi-agent systems for complex product design." Information Systems Frontiers 17, no. 2 (2015): 363-385.

[9] Davoodi, Mohammad Reza, Khashayar Khorasani, Heidar Ali Talebi, and Hamid Reza Momeni. "Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems." Control Systems Technology, IEEE Transactions on 22, no. 3 (2014): 1061-1069.

[10] Wang, Yannan, Yuning Song, and Frank Lewis. "Robust Adaptive Fault-tolerant Control of Multi-agent Systems with Uncertain Non-identical Dynamics and Undetectable Actuation Failures." (2015).

[11] Kumar, Sanjeev, and Philip R. Cohen. "Towards fault-tolerant multi-agent system architecture." Proceedings of the fourth international conference on Autonomous agents. ACM, 2000.

[12] Marin, Olivier, Pierre Sens, Jean-Pierre Briot, and Zahia Guessoum. "Towards adaptive fault tolerance for distributed multi-agent systems." In Proceedings of ERSADS, pp. 195-201. 2001.

[13] Almeida, Alessandro, Jean-Pierre Briot, Samir Aknine, Zahia Guessoum, and Olivier Marin. "Towards autonomic fault-tolerant multi-agent systems." In The 2nd Latin American Autonomic Computing Symposium (LAACS'2007), Petropolis, RJ, Brésil. 2007.

[14] Khan, Zaheer Abbas, Salman Shahid, H. Farooq Ahmad, Arshad Ali, and Hiroki Suguri. "Decentralized architecture for fault tolerant multi agent system." In Autonomous Decentralized Systems, 2005. ISADS 2005. Proceedings, pp. 167-174. IEEE, 2005.

[15] Alessandro de Luna Almeida , Samir Aknine , Jean-Pierre Briot , Jacques Malenfant, Plan-based replication for fault-tolerant multi-agent systems, Proceedings of the 20th international conference on Parallel and distributed processing, p.347-347, April 25-29, 2006, Rhodes Island, Greece.

[16] Singh, Aarti, Dimple Juneja, and A. K. Sharma. "Adaptive and automated fault-tolerance for multi-agent systems." In Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on, vol. 1, pp. 53-57. IEEE, 2011.

[17] Koppensteiner, Gottfried, Munir Merdan, Wilfried Lepuschitz, and Ingo Hegny. "Hybrid based approach for fault tolrance in a multi-agent system." In Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on, pp. 679-684. IEEE, 2009.

[18] Bora, Sebnem, and Oguz Dikenelli. "On the choice of sampling rates in a fault-tolerant multi-agent system." In 2012 International Symposium on Innovations in Intelligent Systems and Applications. 2012.

[19] Mirian, Maryam S., Majid Nili Ahmadabadi, and Zainalabedm Navabi. "A decision-making based approach for fault-handling in multi-agent systems." InNeural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on, vol. 4, pp. 1905-1909. IEEE, 2002.

[20] Khalili, Mohsen, Xiaodong Zhang, Yongcan Cao, and

Jonathan A. Muse. "Distributed Adaptive Fault-Tolerant Consensus Control of Multi-Agent Systems with Actuator Faults".

Authors' Profiles

Yasir Arfat is currently student of M.S in computer science at King Abdulaziz University, Jeddah, Saudi Arabia. He has completed his bachelor degree in Software Engineering with Distinction (Gold Medal) from the University of Azad Jammu & Kashmir, Pakistan in 2011. His research interests include network security, software security, software agent systems and big data, high performance computing.

Fathy E. Eassa received his B.Sc degree in electronics and electrical communication engineering from Cairo University, Egypt in 1978 and the M.Sc. degree in computers and Systems engineering from Al Azhar University, Cairo, Egypt in 1984, and Ph.D degree in computers and systems engineering from Al-Azhar University, Cairo, Egypt with joint supervision with University of Colorado, U.S.A, in 1989. He is a full professor with computer Science dept, Faculty of Computing and Information technology, King Abdul Aziz University, Saudi Arabia. His research interests include agent based software engineering, cloud computing, software engineering, big data, distributed systems, exascale system testing.

How to cite this paper: Yasir Arfat, Fathy Elbouraey Eassa, "A Survey on Fault Tolerant Multi Agent System", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.9, pp.39-48, 2016. DOI: 10.5815/ijitcs.2016.09.06